APPENDIX A.

CLEAN VERSION OF AMENDED CLAIMS

1. A method of loading a trustable operating system comprising:

identifying a region in a memory of a computer by a one of a plurality of processors;

loading a content into the identified region; and

causing the one processor to jump to a known entry point in the content.

2. The method of claim 1, further comprising:

preventing interference with the identifying, loading, and registering by at least one of a remaining one of the plurality of processors.

A2

3. The method of claim 2, wherein preventing interference comprises halting at least one of the remaining ones of the plurality of processors until the identifying, loading, and registering is complete.

4. The method of claim 2, further comprising:

causing at least one of the remaining ones of the plurality of processors to jump to the known entry point in the content.

5. The method of claim 1, wherein identifying comprises receiving a region parameter, the region parameter specifying a location of the region.

6. The method of claim 5, wherein the location comprises a range of addresses in the memory of the computer within which the region is located.

8. The method of claim 1, further comprising:

registering an identity of the content of the identified region, wherein registering comprises:

A3

recording a hash digest of the content of the identified region; and

signing the hash digest, the signed hash digest being stored in a register in the memory of the computer that is accessible to a third party to verify whether the content can be trusted.

32. An apparatus to load a trustable operating system comprising:

a first processor having a start secure operation (SSO), the SSO having a memory region parameter, wherein the first processor is capable of executing the SSO to block access to a region of memory specified in the memory region parameter and to place a content in the specified region;

A4
Cmit

a hash digest, wherein the first processor further is capable of executing the SSO to erase a current content of the hash digest and to record in the hash digest a cryptographic hash of the content of the specified region; and

wherein the first processor further is capable of executing the SSO to unblock access to the specified region and to jump to a known entry point in the content of the specified region.

33. The apparatus of claim 32, further comprising:

a second processor, the second processor having a join secure operation (JSO), wherein the second processor is capable of executing the JSO to prevent the second processor from interfering with the first processor's execution of the SSO.

34. The apparatus of claim 33, wherein the second processor is capable of_ commencing execution of the JSO when the first processor commences execution of the SSO.

35. The apparatus of claim 33, wherein, to prevent the second processor from interfering with the first processor's execution of the SSO, the JSO is capable of causing the second processor to enter a halted state until the first processor's execution of the SSO is complete.

A4 contd.

36. The apparatus of claim 35, wherein the first processor is capable of executing the JSO to further cause the second processor to exit the halted state after the first processor's execution of the SSO is complete and to begin executing at the known entry point in the content of the specified region.

37. The apparatus of claim 32, further comprising a digest signing engine having a secure channel to access the hash digest, the digest signing engine capable of computing the cryptographic hash of the content in the specified region in response to a request by the first processor executing the SSO.